REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE Professional Paper	3. DATES COVERED
4. TITLE AND SUBTITLE Selecting the Most Efficient Reed Solomon Codes to Eliminate Jamming		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Henderson Benjamin, IV		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Air Warfare Center Aircraft Division 22347 Cedar Point Road, Unit #6 Patuxent River, Maryland 20670-1161		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Reed-Solomon (RS) codes are, perhaps, the most widely applied channel codes in practice due to its ability to detect and correct random and burst errors. In today's military combat environment, the ability to communicate determines who wins or losses. For this reason, electronic warfare (EW) and information warfare (IW) have become an area for many discussions and much research. Jamming environments, especially intentional jamming as in military applications, are confronted with the effects of nonrandom errors. The catalog of RS codes is a rather long one. To select a proper code for a given application, the system designer is compelled to deal with numerous tables, graphs, and equations. I have reported the results of designing an artificial neural network (NN) from which one can select the most "efficient" RS code for a specific application. In this article, I present the continuation of my work, in development of an artificial NN for selection of RS codes to eliminate intentional jamming. Student version of the MATLAB Neural Networks Toolbox is used for NN simulation. The Levenberg-Marquardt learning algorithm is used to train the NN. The resultant NN has six inputs, eleven units in the hidden layer, and one unit in the output layer. The output is "k". The test data results show the accuracy of selecting the correct code dimension is 98.04%.

15. SUBJECT TERMS

Reed-Solomon (RS)

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Henderson Benjamin
a. REPORT	b. ABSTRACT	c. THIS PAGE		6	19b. TELEPHONE NUMBER (include area code) (301) 342-0519

SELECTING THE MOST EFFICIENT REED SOLOMON CODES TO ELIMINATE JAMMING (U)

Henderson Benjamin
Naval Air Station Patuxent River, **Benjaminhc@navair.navy.mil**

ABSTRACT

(U) Reed-Solomon (RS) codes are, perhaps, the most widely applied channel codes in practice due to its ability to detect and correct random and burst errors. In today's military combat environment, the ability to communicate determines who wins or losses. For this reason electronic warfare (EW) and information warfare (IW) have become an area for many discussions and much research. Jamming environments, especially intentional jamming as in military applications, are confronted with the effects of non-random errors. The catalog of Reed-Solomon (RS) codes is a rather long one. To select a proper code for a given application, the system designer is compelled to deal with numerous tables, graphs and equations. I have reported the results of designing an artificial neural network (NN) from which one can select the most "efficient" RS code for a specific application [1]. In this article I present the continuation of my work, in development of an artificial NN for selection of RS codes to eliminate intentional jamming. Student version of the MATLAB Neural Networks Toolbox is used for NN simulation. The Levenberg-Marquardt learning algorithm is used to train the NN. The resultant NN has six inputs, eleven units in the hidden layer, and one unit in the output layer. The output is "k". The test data results show the accuracy of selecting the correct code dimension is 98.04%.

I. *Introduction*

(U) Two decades after their first practical applications in Voyager deep space communications system, RS codes have found widespread applications in such diverse areas as satellite communications, compact disk digital audio & Digital Versatile Disk (DVD), HDTV, land mobile communications networks [2], and more recently in ATM networks [3]. RS codes wide acceptance is mostly due to their properties that make them uniquely suitable for error correction in a broad spectrum of applications. Perhaps the single most important practical aspect of RS codes, is their burst error correcting capabilities, which render them attractive for applications in fading channels, jamming environments,and recording systems.

II. *Reed-Solomon Codes*

(U) Reed-Solomon codes considered in this article are exclusively selected from primitive RS codes defined over Galois fields of characteristic 2. This implies that the RS code is a "2^m-ary" linear cyclic code of length $2^m - 1$. A (n, k, t) primitive Reed-Solomon code is generated by;

$$g(x) = \prod_{l=0}^{2t-1} (x + \alpha^{j+l}) \tag{1}$$

where α is a primitive element of GF(2^m), t is the random error correction capability of the code, n is the codeword length, k is the code dimension, and m is the word size. In this article code rate R_c, is considered to be the parameter that defines "efficiency" of the RS code.

(U) Reed-Solomon codes and many modified version of RS codes , such as shortened RS codes, are Maximum Distance Separable (MDS). This implies that, these codes satisfy the Singleton Bound with equality, which implies they provide maximum possible error correction capability relative to number of redundancy symbols that they require [4]. It is well known that RS codes are among the most efficient burst error correcting codes. Reed-Solomon codes guaranteed single, double, and triple burst error correction capabilities are given by the following equations:

$$b_1 = m(t-1)+1 \tag{2}$$

$$b_2 = m[\lfloor \tfrac{t}{2} \rfloor - 1] + 1 \tag{3}$$

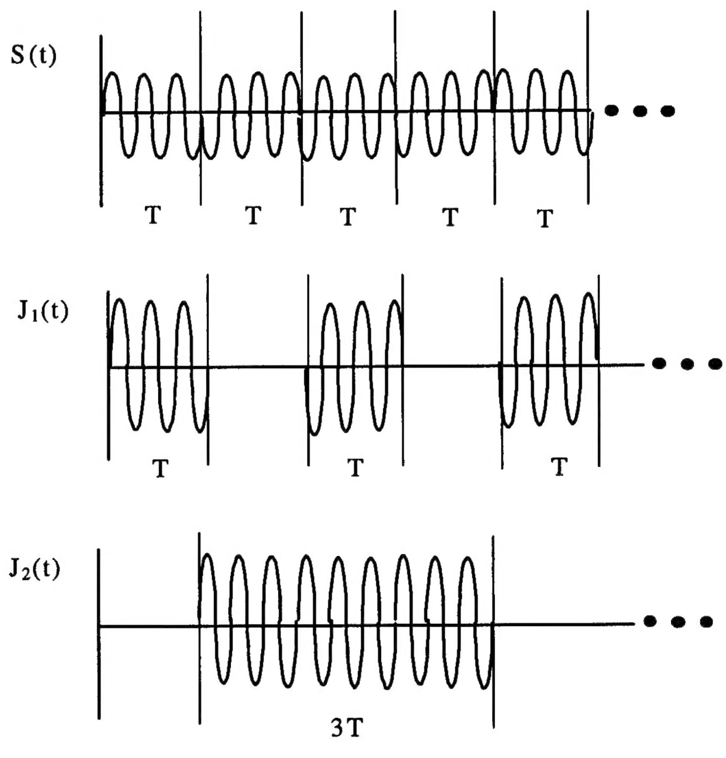
$$b_3 = m[\lfloor \tfrac{t}{3} \rfloor - 1] + 1 \tag{4}$$

Where b_1 , b_2 , and b_3 are single, double, triple burst error correcting capabilities of RS codes [2].

III. *Intentional Jamming*

(U) Jamming environments, especially intentional jamming as in military applications, are confronted with the effects of non-random errors. Communication channels that have non-random characteristics are considered to contain memory and tend to produce what is known as burst errors. In fact, most operational communication system applications exhibit a combination of random and non-random characteristics. These channels are termed diffuse or compound channels [5] and cannot be modeled as AWGN channels. Therefore, in answering the question why burst error correction, most real world communication systems need codes that can combat the common occurrence of burst errors.

(U) In today's military combat environment, the ability to communicate determines who wins or losses. For this reason electronic warfare (EW) and information warfare (IW) have become an area for many discussions and much research. Of particular interest is the survival of a communication system in the presence of intentional jamming. There are many different types of jamming techniques, however pulsed jamming has shown to be the most optimal for creating interference [5]. Pulsed jamming concentrates more of the total signal power into each pulse unlike a continuous wave jammer whose power is more spread out. If the pulse width of the jamming signal is of the same order as the symbol transmission rate, then approximately one symbol will be affected. If the pulse width is greater than the symbol transmission rate, then multiple symbols can be affected thus causing burst errors. As illustrated in figure 1, J1(t) has pulse width approximately equal to the symbol rate of the communication signal S(t). Jamming signal J2(t) has a pulse width approximately equal to three times the symbol rate of S(t) thus causing burst errors.



(U)

Figure 1: Jamming Signal

(U) Techniques such as spread spectrum communication have been employed with good success in defending against intentional jamming. However, it can be shown that using spread spectrum with the addition of error control coding and interleaving techniques, the affect of pulsed noise jamming can almost be eliminated [5]. One example of a military communication system that uses Reed-Solomon codes is the Joint Tactical Information Distribution System (JTIDS) being supported by all branches of the military and NATO [5].

(U) With the ever increasing complexity of communication systems, the powerful burst error correcting Reed-Solomon codes will no doubt see more applications in digital recording systems, mobile communication systems, and in communication environments with interference.

(U) Another important interleaving technique worth mentioning is called pseudorandom interleaving. Pseudorandom interleavers work on the principle of using synchronous pseudorandom number generators in both the interleaver and deinterleaver. The pseudorandom codes are used to vary the depth of interleaving in a manner only known to the communication system. The importance of this technique is in combating intentional jamming in hostile communication environments. If the depth of an interleaver is known, a pulsed jammer can be used with a pulse repetition interval (pri) equal to the interleaving depth in order to produce burst errors at the output of the deinterleaver [5]. By using pseudorandom interleaving depths pulsed jammers lose their effectiveness.

(U) The number of Reed-Solomon codes in existence are rather large when considering codes can be modified (shorten, lengthen, and puncturing), interleaved, and concatenated. The designer is faced with numerous tables, charts and graphs when selecting codes. The research work accomplished with Neural Networks proves that the same method can be helpful in eliminating the effects of jamming.

IV. *Neural Networks*

(U) A NN has the ability to derive meaning from complicated or imprecise data, extract patterns, and detect trends that are too complex for humans to recognize by any other computing technique. A NN uses a training rule where the weights and biases of the connections adjust for inputs based on the outcome. A trained NN is an expert on the information it analyzes. One of the inherent strengths of a NN is its ability to forecast or predict an outcome. Therefore, a NN can select a shortened Reed-Solomon code if properly trained. The results prove that a NN could select shortened Reed-Solomon codes for a particular application.

(U) The MATLAB Neural Network Toolbox was used to perform the software simulation. It contains all the necessary functions for generating the Levenberg-Marquardt (LM) algorithm. The default data for the LM algorithm was used with the exception of the number of epochs to train and the sum squared error (SSE) goal. Software development was required for generating training and test data, pre-processing data into the NN, and post-processing of data coming out of the NN.

V. *Designing the Neural Network*

(U) The function of the designed is to select the “best” RS code from a large pool, for a set of code parameters derived from a requirements application. The NN makes its decision on the basis of selecting the code possessing the most efficient code rate.

A NN was developed select RS code of codeword lengths 7, 15, 31, 63, 127, and 255. The design of a NN involves determining the type of algorithm for learning, gathering training and test data, determining the architecture, adequately training the NN, and then testing the NN. The LM method is chosen as the algorithm.

(U) The main code parameters of interest to the designer are assumed to be symbol length m, random error correcting capability t, code rate R_C, and single, double, and triple burst error correction capabilities. These parameters are considered to be potential design requirements and represent the patterns to be presented to the NN. These parameter are considered the requirements of a designer and represent the patterns to be presented to the NN. The symbol length and any combination of the remaining parameters could be presented to the network. If the symbol length is not present, the most efficient Reed-Solomon code would always result in a 255 class length being selected.

V. *Training the Neural Network*

(U) From a pool of unmodified uninterleaved Reed-Solomon codes, 125 Reed-Solomon codes were selected to generate patterns for the training data. Then a set of 92 Reed-Solomon codes different from the training codes were selected as test cases. The number of vector patterns for each code varies between two and six vectors for each of the codes for the training and test. The amount of data varies because some codes contain less data. The training process, utilizing the LM method, contained an architecture with five inputs, nine units in the hidden layer, and two units in the output layer. The NN was trained for 1500 epochs which resulted in a sum squared error of 5.589×10^{-4}. An epoch represents the entire training data examples. The network was trained with 685 examples containing various parameters of RS codes.

VII. *Evaluation of Data*

(U) The test data represent data that was never seen by the NN. The training data allows the network to generalize or forecast patterns that it has never seen. The bold faced pattern designates an incorrect selection by the NN. The total number of patterns used during the testing process 510. The test data results showed the accuracy of selecting the correct code length and dimension is 98.4% [6].

VIII. *Conclusion*

(U) The methodology used in selecting the best Reed Solomon code using a Neural Network can also select the appropriate code for military application. With the use of interleaving of Reed-Solomon codes, spread-spectrum or frequency hopping technology, and the Neural Network described in this article can eliminate the effects of intentional jamming.

IX. *References*

[1] H. C. Benjamin & B. Kamali, "Selection of the Most "Efficient" Reed-Solomon Code for a Specific Application Using Neural Networks," Communications Theory Mini Conference 99, Proceedings of International Conference in Communications (ICC'99), Vancouver B. C. Canada, May 1999.

[2] B. Kamali, "Some New Outlooks on Burst Error Correcting Capabilities of Reed-Solomon Codes with Applications in Mobil Communications," Proceedings of IEEE Vehicular Technology Conference Ottawa, Canada, May 1998.

[3] A. Almulhem & et. al, " Adaptive Error Correction for ATM Communications Using Reed-Solomon Codes," Proceeding of IEEE Southeastern Conf., April 1996.

[4] S. B. Wicker, "Error Control Systems for Digital Communication and Storage," Prentice-Hall, 1995.

[5] R. L. Brinkley, "Burst Error Correction with Reed-Solomon Codes," Master's Thesis, Mercer University, May 1998.

[6] H. Benjamin, "Neural Network System that Selects Reed-Solomon Codes for a Specific Application," Master's Thesis, Mercer University, Mat 1998.